

REMARKS

Subject to entry of the amendments herein, claims 1-32 are pending in the application. By this Amendment Applicant has amended claims 1 and 18 and added new claim 29.

Claim Rejections Under 35 U.S.C. §103(a)

In the first rejection made within the above Office Action, the Examiner rejected claims 1-31 under 35 U.S.C. §103(a) as being unpatentable over Dietz (U.S. Pat. No. 6,665,725) in view of Schweitzer (U.S. Pat. App. Pub. 2002/0016843).

Dietz System

Dietz describes a method of performing protocol specific operations on a packet passing through a connection point on a computer network. The packet contents conform to protocols of a layered model wherein the protocol at a particular layer level may include one or a set of child protocols defined for that level. According to one implementation of the Dietz system, a network packet monitor examines the packets passing in either direction past its connection point and can elucidate what application programs are associated with each packet. The network packet monitor of Dietz is described as follows:

FIG. 3 shows a network packet monitor 300, in an embodiment of the present invention that can be implemented with computer hardware and/or software. The system 300 is similar to monitor 108 in FIG. 1. A packet 302 is examined, e.g., from a packet acquisition device at the location 121 in network 102 (FIG. 1), and the packet evaluated, for example in an attempt to determine its characteristics, e.g., all the protocol information in a multilevel model, including what server application produced the packet.

The packet acquisition device is a common interface that converts the physical signals and then decodes them into bits, and into packets, in accordance with the particular network (Ethernet, frame relay, ATM, etc.). The acquisition device indicates to the monitor 108 the type of network of the acquired packet or packets.

Aspects shown here include: (1) the initialization of the monitor to generate what operations need to occur on packets of different types--accomplished by compiler and optimizer 310; (2) the processing--parsing and extraction of selected portions--of packets to generate an identifying signature--accomplished by parser subsystem 301, and

(3) the analysis of the packets--accomplished by analyzer 303.
[8:48-9:3]

As described by Dietz, one operation effected by the analyzer 303 is to look up records of known flows based upon a parser record provided by the parser 301:

The parser record is passed onto lookup process 314 which looks in an internal data store of records of known flows that the system has already encountered, and decides (in 316) whether or not this particular packet belongs to a known flow as indicated by the presence of a flow-entry matching this flow in a database of known flows 324. A record in database 324 is associated with each encountered flow.

[10:58-65]

That is, the parser 301 passes records corresponding to each type of flow processed by the packet monitor 300 to the analyzer subsystem 303. It is only within the analyzer subsystem 303 itself that a protocol and state identification process 318 determines the protocol characterizing the flow:

If there is no flow-entry found matching the signature, i.e., the signature is for a new flow, then a protocol and state identification process 318 further determines the state and protocol. That is, process 318 determines the protocols and where in the state sequence for a flow for this protocol's this packet belongs. Identification process 318 uses the extracted information and makes reference to the database 326 of state patterns and processes. Process 318 is then followed by any state operations that need to be executed on this packet by a state processor 328.

[11:41-50]

As is discussed below, this aspect of Dietz is in direct contravention to aspects of the invention as presently claimed; namely, that an event type or flow protocol determined to be associated with a received event is processed by a particular processing core known to be compatible with the event type or flow protocol. Since in the Dietz system the particular flow and protocol associated with a given packet is not determined until *after* such packet is passed to the analyzer subsystem 303, it follows that Dietz teach away from the *assignment* of a specific analyzer known to be compatible with a protocol or event type to a flow determined to be of a given protocol or to contain an event of such event type. This is true even to the extent that the packet monitor of Dietz is capable of being configured to execute multiple analyzers in parallel, since this in no way changes the fact that the parser of Dietz's system would still supply the parallel analyzers with parser records of input packets independent of the flow/protocol corresponding to the packet. That is, Dietz's system is incompatible with any system which would contemplate an assignment of analyzers known to be compatible with flows determined to

be of particular types, since flow type and protocol is determined within the analyzer itself and not beforehand.

Schweitzer System

In the above Office Action, the Examiner has characterized the system described by Schweitzer as follows:

Schweitzer discloses a method and system in which packets of a flow are stored in a queue and examined to determine an application associated with a flow according to protocol (see paragraphs 0034-0037). Moreover, packets are initially processed by a filter for performing protocol recognition and then assigned to a packet analyzer. Most importantly, Schweitzer discloses using multiple packet analyzers and assigning flows to appropriate one of the analyzers based upon the protocol filter criteria (see paragraphs 0034-0037).

Applicant respectfully submits that, contrary to the Examiner's contention, Schweitzer does not describe or suggest the assignment of particular flows to analyzers, or vice-versa. In this regard the Examiner cites the following paragraphs of Schweitzer in support of this contention:

[0034] The following describes the uses of the elements of FIG. 1. The packet sources 100a-e could be network connections, local computers, network computers, the Internet, and/or some other type of packet source. The packet sources 100a-e are sources of packets such as IP packets, IPX packets, and/or some other type of packets.

[0035] In some embodiments the filter 102 is provided to filter out packets. In other embodiments, no filter 102 is used. The filter 102 can be set to remove local traffic from further analysis, e.g. packets not leaving the corporate Intranet, or packets not travelling over a particular backbone. Additionally, if multiple analyzers like the analyzer 104 are being used, then multiple filters like the filter 102 can be used to segment the analysis. For example, all voice over IP calls might be filtered out by one filter but be the only thing passed through by another. This allows for tremendous flexibility in providing distributed analysis and meaningful analysis. In some embodiments, a standard packet capture (pcap) language is used to define the filter, e.g. "tcp and port 80 or dst net 192.168.0.0 mask 255.255.0.0", etc.

[0036] *Only those packets that meet the tests of the filter 102 are passed to the analyzer 104.* In some embodiments, the filter 102 and the analyzer 104 are hosted on separate computers. For example, the filter 102 might have two Ethernet interfaces, one for receiving packets from the packet sources 100a-e and the other for sending matching packets to the analyzer 104.

[0037] Packets are analyzed by the analyzer 104 to be assigned to flows and then to sessions. The analyzer 104 can gather statistics about flows and sessions for use by the data collector 106. Each of the components of the analyzer 104 can be performed on a single computer and/or multiple computers to support distributed processing.
[italics added]

As indicated by the foregoing excerpts, Schweitzer's filter does not function to assign events or flows to particular analyzers based upon event type or stateful protocol criteria; rather, Schweitzer's filter criteria (e.g., address ranges, port numbers) does not pertain to event type or stateful protocol. For example, Schweitzer describes certain embodiments in which no filter may be used, and other embodiments in which the applicable filter criteria is set to prevent local traffic from being further analyzed (e.g., by filtering as a function of address range). In any case, it is clear that *Schweitzer's filter is not configured to recognize flows* and make corresponding assignments to analyzers, or vice versa. Rather, in Schweitzer's system the flow type of a packet is not determined until *after* the packet has been provided to an analyzer, which means that Schweitzer actually teaches away from assigning events or flows to protocol processors as presently claimed:

[0037] *Packets are analyzed by the analyzer 104 to be assigned to flows* and then to sessions. The analyzer 104 can gather statistics about flows and sessions for use by the data collector 106.
[emphasis added]

It follows that since Schweitzer's filter is not equipped to identify or otherwise recognize particular flows and since flow recognition occurs within Schweitzer's analyzer, Schweitzer in fact teaches away from the assignment of flows to analyzers known to be compatible with particular stateful protocols or event types based upon protocol filter criteria.

Differences Between the Cited References and the System of the Invention

As defined by the pending claims, the present invention pertains in one aspect to a system in which different protocol processing cores known to be compatible with a particular stateful protocol or event type are assigned to process particular portions of a given flow determined to be characterized by the protocol or to contain events of the event type.

As discussed in the preceding sections, neither the Dietz nor Schweitzer reference, either alone or in combination, describe various aspects of the invention as presently claimed. For example, neither of these references describe at least the following aspects of claim 1:

(1) *The deriving of an event of a particular type where the event is associated with a flow comporting with a particular stateful protocol.* In this regard the Examiner contends that Dietz discloses “type A and type B messages”, and cites col. 7 lines 12-26 and col. 9, lines 9-27 in support of this contention. However, Neither Dietz nor Schweitzer describes deriving events of particular types that are associated with a particular stateful protocol. In particular, the “type A” and “type B “ events do not in any way relate to a stateful protocol, but instead are associated with an application:

For example, in a conversational flow associated with a particular application, the application will cause the server to send a type-A packet, but so will another. If, though, the particular application program always follows a type-A packet with the sending of a type-B packet, and the other application program does not, then in order to recognize packets of that application's conversational flow, the monitor can be available to recognize packets that match the type-B packet to associate with the type-A packet.

(7:15-23)(emphasis added)

The other portion of Dietz referenced by the Examiner, i.e., col. 9, lines 9-27, has no relevance to the application-level “type-A” and “type-B” packets referenced in the excerpt above. It follows that although Dietz describes that different types of packets may be associated with an application, this in no way describes or suggests the deriving events of particular types comporting with a stateful protocol as presently claimed.

(2) *The assignment of a first protocol processing core to process events of a first type derived from a flow comporting with a stateful protocol and the assignment of a second protocol processing core to process events of a second type derived from the flow.* In this regard the Examiner observes that “Dietz discloses the use of various protocols for transmitting data through a network” and that “Dietz discloses the use of different protocols and application stateful protocols such as TCP/IP”. However, neither of these statements in any way explain how Dietz describes the claim limitations referenced above. Specifically, even if Dietz does in fact support the use of different

protocols to transmit data through a network, this in and of itself does not describe or suggest the assignment of particular protocol processing cores to process different types of events derived from a single flow. Although certain of the portions of Dietz cited by the Examiner relate to flows, i.e., col. 9, lines 9-27 and col. 13, lines 5-25, these portions do not pertain to the assignment of packets, flows or events to particular analyzers. The remaining portion of Dietz cited by the Examiner, i.e., col. 6, lines 6-31, is also not related to the assignment of packets or flows to particular processing cores. Moreover, this excerpt is also directed to application-level processing and is thus completely inapposite to the assignment of processing cores at the flow-level. Accordingly, Applicant respectfully submits that Dietz fails to describe or suggest the presently claimed assignment of different protocol processing cores to process events of different types derived from a single flow.

(3) *The selection of a protocol processing core from among a plurality of protocol processing cores compatible with an event type.* As mentioned above, in Schweitzer's system the flow type of a packet is not determined until after the packet has been provided to an analyzer, which means that Schweitzer actually teaches away from assigning events or flows to protocol processors as presently claimed.

Notwithstanding these distinctions between the invention as presently claimed and the cited references, in order to advance prosecution of the application claims 1 and 18 have been amended to recite that the first protocol processing core is selected from among a plurality of protocol processing cores identified in a memory structure as being compatible with events of the first type. As has been previously discussed, neither Dietz nor Schweitzer describe the assignment of packets, flows or events determined to be of a particular type to particular analyzers or processing cores compatible with such type. The present amendments further distinguish the pending claims from the cited references by explicitly reciting that specific identification of those processing cores compatible with various event types is maintained within a memory structure. See, e.g., the present specification at page 54, lines 5-9. Neither Dietz nor Schweitzer describe or suggest establishing such a memory structure. This is to be expected

given that neither Dietz nor Schweitzer describes the claimed assignment of packets, events or flows to processing cores or analyzers determined to be compatible therewith.

Accordingly, Applicant respectfully requests reconsideration of the outstanding rejection of claims 1-28 under 35 U.S.C. §103(a) in view of the above amendments and remarks.

With regard to the outstanding rejection of claims 29-31, Applicant observes that the Examiner has stated:

Claims 18-31 do not teach above and beyond the limitations of claims 1-17 and therefore are rejected under the same rationale.

Although certain similarities exist between claims 1-17 and claims 18-31, Applicant observes that claims 18-31 are apparatus claims which include elements not explicitly recited by method claims 1-17. With respect to claims 29-31, Applicant respectfully submits that the cited references do not include structures corresponding to, for example, the following claim elements:

- first/second clusters of protocol processing cores including first/second pluralities of protocol processing cores compatible with a first/second stateful protocols;
- a dispatcher;
- a scratchpad memory operatively connected to said dispatcher;
- a socket memory controller operatively connected to said scratchpad memory.

Accordingly, Applicant respectfully requests reconsideration of the outstanding rejection of claims 29-31.

The Examiner has also provisionally rejected claims 1-31 on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-41 of copending Application No. 10/211,434. Upon receiving an indication that the pending claims are otherwise allowable, Applicant file will a terminal disclaimer to overcome this provisional rejection.

Support for new claim 32 can be found in the specification at, for example, page 54, lines 5-9.

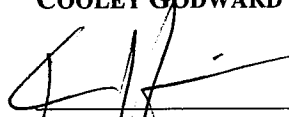
Applicant respectfully requests consideration of the remarks herein prior to further examination of the above-identified application. The undersigned would of course be available to discuss the present application with the Examiner if, in the opinion of the Examiner, such a discussion could lead to resolution of any outstanding issues.

Dated: November 2, 2007

COOLEY GODWARD KRONISH LLP
ATTN: Patent Group
777 6th Street NW, Suite 1100
Washington, DC 20001

Respectfully submitted,
COOLEY GODWARD KRONISH LLP

By:


Kevin J. Zimmer
Reg. No. 36,977

Tel: (858) 550-6000
Fax: (202) 842-7899